An Enhanced Fuzzy Single Layer Perceptron for Image Recognition

Jong-Hee Lee†

ABSTRACT

In this paper, a method of improving the learning time and convergence rate is proposed to exploit the advantages of artificial neural networks and fuzzy theory to neuron structure. This method is applied to the XOR problem, n bit parity problem which is used as the benchmark in neural network structure, and recognition of digit image in the vehicle plate image for practical image application. As a result of the experiments, it does not always guarantee the convergence. However, the network showed improved the learning time and has the high convergence rate. The proposed network can be extended to an arbitrary layer. Though a single layer structure is considered, the proposed method has a capability of high speed learning even on large images.

이미지 인식을 위한 개선된 퍼지 단층 퍼셉트론

이 종 희†

요 약

본 논문에서는 인공 신경망과 퍼지 논리의 장점을 뉴런 구조에 적용하여 학습 속도가 빠르며 수렴률을 향상시키는 방법을 제안한다. 인공 신경망의 벤치 마크로 사용되는 XOR문제 n 비트 parity문제와 현실적인 이미지 응용을 위해 자동차 번호판에서 숫자 이미지에 적용시켜 보았다. 실험 결과, 모든 자료값과 목표값에 대해서 항상 수렴을 보장하는 것은 아니다. 그렇지만, 학습 속도가 빠르며 수렴률의 향상을 보였다. 제안된 방법은 임의의 층으로 확장이 가능하다. 여기서는 단층의 경우만을 고려하여 빠른 속도와 방대한 이미지에 대해서 빠른 처리를 가능하게 한다.

1. Introduction

In recent image recognition researchs, there is a vigorous interest to apply artificial networks and fuzzy theory not only on simple image recognition but also on complex applications. However, we may face the following problems-oscillation at local minima, high price in learning speed, misidentification to degrade the efficiency of recognition etc. as well as inability to regulate rules using respective existing artificial networks and fuzzy theory.

In the conventional single layer perceptron, it is inappropriate to use when a decision boundary for classifying input patterns does not composed of hyperplane. Moreover, the conventional single layer perceptron, due to its use of unit function, was highly sensitive to change in weights, difficult to be implemented and could not learn from past data[1]. Therefore, it could not find a solution of the exclusive OR problem, the benchmark.

There are a lot of endeavor to implement a fuzzy theory to artificial neural network[2]. Goh et al.[3] proposed the fuzzy single layer perceptron algorithm,

† 신라대학교 컴퓨터정보공학부

and advanced fuzzy perceptron based on the generalized delta rule to solve the XOR problem, and the classical problem[3]. This algorithm guarantees some degree of stability and convergency in application using fuzzy data. However, it causes an increased amount of computation and some difficulties in application of the complicated image recognition. The enhanced fuzzy perceptron has shortcomings such as the possibility of falling in a local minima and slow learning time[4].

In this paper, I propose a enhanced fuzzy single layer learning algorithm. I construct and train a new type of fuzzy neural net to model the linear function. Properties of this new type of fuzzy neural net include : (1) proposed linear activation function; and (2) a modified delta rule for learning. I will show that such properties can guarantee to find solutions for the problems-such as exclusive OR, 3-bits parity, 4-bits parity and digit image recognition on which simple perceptron and simple fuzzy perceptron can not.

2. Fuzzy Single Layer Perceptron

Wang proposed a fuzzy single layer perceptron using the algorithm based on generalized delta rule[3]. This algorithm guarantees some degree of stability and convergency in application of data. However it causes an increased amount of computation and some difficulties in application of the complicated pattern recognition.

2.1 Learning Algorithm for the Two Input Fuzzy Single Layer Perceptron

For each input, repeat step one and two while error is minimized,

Step 1 : Sort the input of the perceptron such that $I_1 \leq I_2$.

The output of the network can be expressed as follows:

$$Output = I_1 * f(W_1 + W_2) + (I_2 - I_1) \times f(W_2)$$

Where $f(W) = 0 \quad W < 1$
$f(W) = 1 \quad W \geq 1$
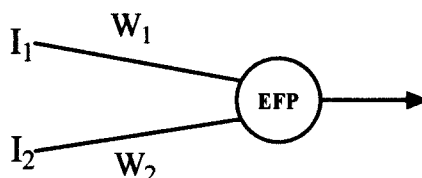
Fig.1 The two input single layer fuzzy perceptron

The expression may be viewed as a conventional neural network.
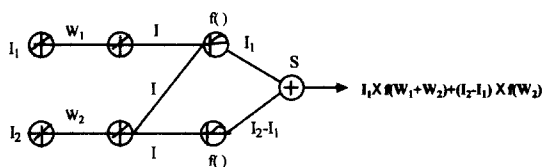
Fig.2 The two input single layer fuzzy perceptron architecture

Step 2 : Apply the generalized delta rule to the above sorted network, I derive the incremental changes for the weights, $\triangle W_1$ and $\triangle W_2$.

$$\triangle W_1 = \varepsilon * I_1 \times f(W_1 + W_2)$$
$$\triangle W_2 = \varepsilon * I_1 \times f(W_1 + W_2) + \varepsilon \times (I_2 - I_1) \times f(W_2)$$

3. An Enhanced Fuzzy Single Layer Perceptron

In this section the learning algorithm for a single layer perceptron is proposed. Before I discuss the new learning algorithm, we introduce the proposed learning architecture. Figure 3 shows the architecture of the new learning algorithm.

3.1 An Enhanced Fuzzy Single Layer Learning Algorithm

A proposed learning algorithm can be simplified and divided into four steps. For each input, repeat step 1, step 2, step 3, and step 4 while error is minimized
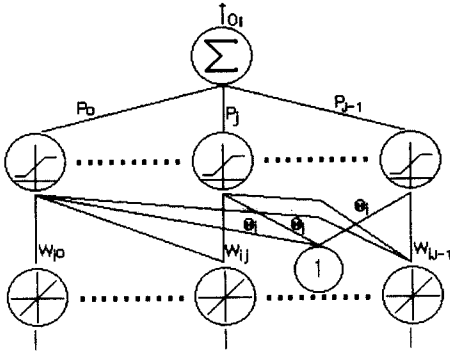
Fig.3 A proposed fuzzy single layer perceptron model

Step 1 : Initialize weights and bias terms.

Define W_{ij}, $(1 \le i \le I)$, to be the weight from input j to output i at time t, and θ_i to be a bias term in the output soma. Set $W_{ij}(0)$ to small random values, thus initializing all the weights and bias term.

Step 2 : Rearrange A_i in ascending order of membership degree m_j and add an item m_0 at the beginning of this sequence.

$$0.0 = m_0 \le m_1 \le \ldots \le m_j \le m_J \le 1.0$$

Compute the consecutive difference between the items of the sequence.

$$P_k = m_j - m_{j-1}$$
$$\text{Where } k = 0, \ldots, n$$

Step 3 : Calculate a soma (O_i)'s actual output.

$$O_i = \sum_{k=0}^{I-1} P_k * f(\sum_{j=k}^{I-1} W_{ij} + \theta_i)$$

Where $f(\sum_{j=k}^{I-1} W_{ij} + \theta_i)$ *is linear activation function*
Where $i = 1, \ldots, I$.

In the sigmoid function, if the value of $(\frac{1.0}{1.0+e^{-net}})$ is between 0.0 and 0.25, $((\frac{1.0}{1.0+e^{-net}}) * (1-\frac{1.0}{1.0+e^{-net}}))$ is very similar to $(\frac{1.0}{1.0+e^{-net}})$. If the value of $(\frac{1.0}{1.0+e^{-net}})$ is between 0.25 and 0.75, $((\frac{1.0}{1.0+e^{-net}}) * (1-\frac{1.0}{1.0+e^{-net}}))$ is very similar to 0.25. If the value of $(\frac{1.0}{1.0+e^{-net}})$ is between 0.75 and 1.0, $((\frac{1.0}{1.0+e^{-net}}) * (1-\frac{1.0}{1.0+e^{-net}}))$ is very similar to $(1-\frac{1.0}{1.0+e^{-net}})$.

Therefore, the proposed linear activation function expression is represented as follows : The formulation of the activation linear function is following.

$$f(\sum_{j=k}^{I-1} W_{ij} + \theta_i) = 1.0 \quad where(\sum_{j=k}^{I-1} W_{ij} + \theta_i) > 5.0$$
$$f(\sum_{j=k}^{I-1} W_{ij} + \theta_i) = \rho * (\sum_{j=k}^{I-1} W_{ij} + \theta_i) + 0.5$$
$$where -5.0 \le (\sum_{j=k}^{I-1} W_{ij} + \theta_i) \le 5.0, \ \rho \in [0.1, 0.4]$$
$$f(\sum_{j=k}^{I-1} W_{ij} + \theta_i) = 0.0 \quad where(\sum_{j=k}^{I-1} W_{ij} + \theta_i) < -5.0$$
$$f((\sum_{j=k}^{I-1} W_{ij} + \theta_i)) = (\frac{1}{range*2}) * (\sum_{j=k}^{I-1} W_{ij} + \theta_i) + 0.5$$

where the range means monotonic increasing interval except for the interval between 0.0 and 1.0 of value of the $f(\sum_{j=k}^{I-1} W_{ij} + \theta_i)$.

Step 4 : Apply the modified delta rule. And I derive the incremental changes for weight and bias term.

$$\triangle W_{ij}(t+1) = \eta_i * E_i * \sum_{k=0}^{i} P_k * f(\sum_{j=k}^{I-1} W_{ij} + \theta_i) + \alpha_i * \triangle W_{ij}(t)$$
$$W_{ij}(t+1) = W_{ij}(t) + \triangle W_{ij}(t+1)$$
$$\triangle \theta_i(t+1) = \eta * E_i * f(\theta_i) + \alpha_i * \triangle \theta_i(t)$$
$$\theta_{ij}(t+1) = \theta_{ij}(t) + \triangle \theta_{ij}(t+1)$$

where η_i *is learning rate* α_i *is momentum.*

Finally, I enhance the training speed by using the dynamical learning rate and momentum based on the division of soma.

if $(Inactivation_{totalsoma} - Activation_{totalsoma} > 0)$ *then*
  $\triangle \eta_i(t+1) = E^2$
  $\eta_i(t+1) = \eta_i(t) + \triangle \eta_i(t+1)$
*end*if
if $(Inactivation_{totalsoma} - Activation_{totalsoma} > 0)$ *then*
  $\triangle \alpha_i(t+1) = E^2$
  $\alpha_i(t+1) = \alpha_i(t) + \triangle \alpha_i(t+1)$
*end*if

3.2 Error criteria problem by division of soma

In a conventional learning method, learning is continued until squared sum of error is smaller than error criteria. However, this method is contradictory to the physiological neuron structure and takes place the occasion in which a certain soma's output is not any longer decreased and learn no more[5]. The error criteria was divided into activation and inactivation criteria. One is an activation soma's criterion of output "1", the other is an inactivation soma's of output "0". The activation criterion is decided by soma of value "1" in the discriminant problem of actual output patterns, which means in physiological analysis that the pattern is classified by the activated soma. In this case, the network must be activated by activated somas. The criterion of activated soma can be set to the range of [0,0.1].

In this paper, however, the error criterion of activated soma was established as 0.05. On the other hand, the error criterion of inactivation soma was defined as the squared error sum, the difference between output and target value of the soma. The degree of activation and inactivation was equally set up on the basis that activation and inactivation is the same[5,6]. Fig.4 shows the proposed algorithm.

```
while ((Activation_no == Target_activated_no)
      &&(Inactivation_error <= Inactivation_area))
     do {
        for (i=0; i<Pattern_no; i++)
           for(j=0; j<Out_cell_no;j++) {
              Forward Pass;
              Backward Pass;
              if(Out_cell==Activation_soma&&|error|<=
                    Activation_area)
                 Activation_number++;
              if (Out_cell==Inactivation_soma)
                 Inactivation_error += error * error;
           }
     }
```

Fig.4 Learning Algorithm by division of soma

4. Simulation & Result

I simulated the proposed method on IBM PC/586 in C++ language. In order to evaluate the proposed algorithm, I applied it to the exclusive OR, 3-bits parity which is a benchmark in neural network and Pattern recognition problems, and a kind of image recognition problem. In the proposed algorithm, the error criteria of activation and inactivation for soma was set to 0.09.

4.1 Exclusive OR and 3-bits parity problems

Here, I set up initial learning rate and initial momentum as 0.5 and 0.75, respectively. Also I set up the range of weight [0,1]. In general, the range of weights were [-0.5,0.5] or [-1,1]. In Table 1, I compare a fuzzy single layer perecptron with the proposed algorithm and all data are the average of 10 runs.

As shown in Table 1 and Table 2, The proposed method showed higher performance than fuzzy perceptron in convergence epochs and convergence rates of the three tasks.

Table 1. Convergence rate in initial weight range

Applied Problem	Fuzzy Perceptron	Proposed Algorithm	Initial Weight Range
exclusive OR	100%	100%	[0.0, 1.0]
	89%	99%	[0.0, 5.0]
3 bit parity	83%	97%	[0.0, 1.0]
	52%	96%	[0.0, 5.0]
4 bit parity	0%	96%	[0.0, 1.0]
	0%	95%	[0.0, 5.0]

Table 2. Comparison of step numbers

Benchmark	Fuzzy Perceptron	Proposed Algorithm
Exclusive OR	8 (converge)	4 (converge)
3 bit parity	13 (converge)	8 (converge)
4 bit Parity	0 (not converge)	15 (converge)

4.2 Digit recognition test in vehicle plate images

I extracted digit images in a vehicle plate using the method in [7]. I extracted the license plate from the image using the characteristic that green color area of the license plate is denser than other colors. The procedure of image pre-processing is presented in Fig.5 and the digit images I used are shown in Fig.6.
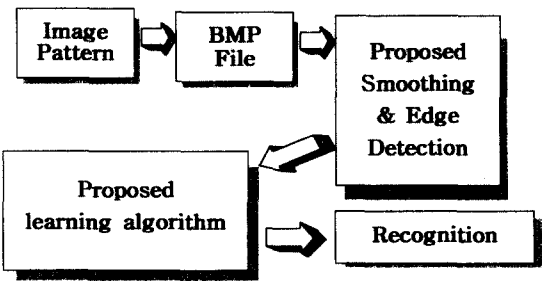
Fig.5 Preprocessing diagram
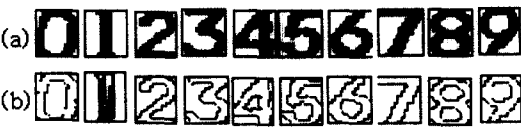
Fig.6 (a) Digit Images and (b) training images by edge detection

I carried out image pre-processing in order to prevent high computational load as well as loss of information. If the extracted digit images were used as training patterns, it requires expensive computational load. In contrast, skeleton method causes loss of important information of images. To overcome this trade-off, I used edge information of images.

The most frequent value method I had developed was used for image pre-processing. This method was used because blurring of boundary using common smoothing methods. Thus, it degrades both color and contour lines[8,9]. The new method replaced a pixel's value with the most frequent value among specific neighboring pixels. If the difference of absolute value between neighborhoods is zero in a given area, the area was considered as background.

Otherwise, it was considered as a contour. This contour was used as a training pattern.

The input units were composed of 32 × 32 array for image patterns. In simulation, the fuzzy perceptron was not converged, but the proposed method was converged on 70 step at image patterns. Table 3 shows the summary of the results in training epochs between two algorithms.

Table 3. The comparison of epoch number

Image Pattern	Epoch Number
fuzzy perceptron	0 (not converge)
proposed algorithm	70 (converge)

5. Conclusions

The study and application of fusing fuzzy theory with logic and inference and neural network with learning ability has been actually achieving according to expansion of automatic system and information processing, etc.

I have proposed a fuzzy supervised learning algorithm which has greater stability and functional varieties compared with the conventional fuzzy perceptron.

The proposed network is able to extend the arbitrary layers and has high convergence in case of two layers or more. Though I considered only the case of the single layer, the networks had the capability of high speed during the learning process and rapid processing on huge image patterns. The proposed algorithm shows the possibility of the application to the image recognition besides benchmark test in neural network by single layer structure.

In future study, I will develop a novel fuzzy learning algorithm and apply it to the face recognition.

References

[1] F. Rosenblatt, "The perceptron : A perceiving and recognizing automaton," Cornell Univ., Ithaca, NY, Project PARA Cornell Aeronaut Lab, Rep., 85-460-1, 1957.

[2] M. M. Gupta and J. Qi, "On Fuzzy Neuron Models," IJCNN, Vol.2, pp.431-435, 1991.

[3] TH Goh, PZ Wang, and HC Lui, "Learning Algorithm for Enhanced Fuzzy Perceptron," IJCNN. Vol2, pp.435-440, 1992.

[4] K. B. Kim, E. Y. Cha, "A New Single Layer Perceptron using Fuzzy Neural Controller," Simulators International XII, Vol.27, No.3, pp.341-343, 1995.

[5] K. B. Kim, J. H. Lee, and E. Y. Cha, "The Fuzzy Neuron Learning Algoritm Based on The Ohysiological Organization," Proceedings of ITC-CSCC'96, Vol.1, pp.345-348, 1996.

[6] Judith E. Dayhoff, "Biological Synapse," Neural Network Architectures, pp.136-162, 1989.

[7] Y. K. Lim, K. B. Kim, "Recognition System of a Car License Plate using Color Information," Proceedings of the Korea Multimedia Society, Vol.2, No.2, pp.377-381, 1999.

[8] D. G. Lowe, "Organization of Smooth Image Curves at Multiple Scales," International journal of computer vision, 1:119-130, 1989.

[9] R. B. Paranjape, R. N. Rangayyan, W. M. Morrow, and H. N. Nguyen, Adaptive neighborhood image processing, CVGIP Graphical Models and Image Processing, 54(3):259-267, 1992.

이 종 희

1978년 경북대학교 공과대학 전자공학과 졸업(공학사)
1984년 경북대학교 대학원 전자공학과 전산공학 졸업(공학석사)
1990년 경북대학교 대학원 전자공학과 전산공학전공 졸업(공학박사)
1979년~1987년 경남정보대학 조교수
1988년~현재 신라대학교 컴퓨터정보공학부 교수
관심분야 : 인터넷응용, 퍼지신경망, 컴퓨터교육
e-mail : jhlee@silla.ac.kr